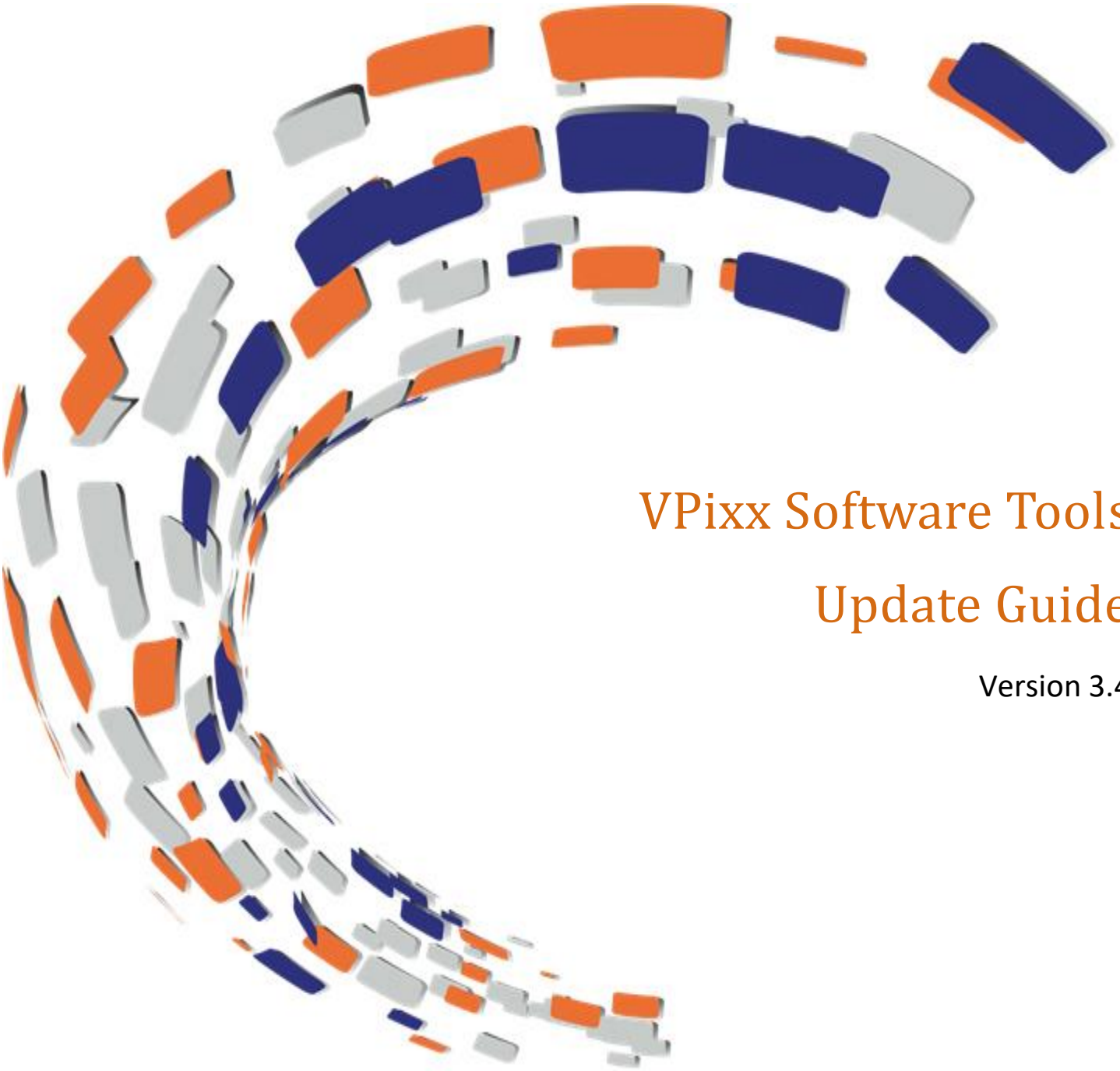


VPixx Software Tools

Update Guide

Version 3.4

Table of Contents

Table of Contents 1
Overview 2
New Firmware Updates 3
PROPixx, revision 41 3
VIEWPixx / PROPixx Controller revision 51 3
TRACKPixx3, revision 11 3
DATAPixx3, revision 12 3
New Software Features – Eye Tracking 4
Pupil size calibration (TRACKPixx3 only) 4
PyPixx Implementation 4
Calibration Validation Script (Matlab) (TRACKPixx3 and TRACKPixx /mini) 7
TRACKPixx /mini data access new function and speed improvement 8
Blink detection digital output mode (TRACKPixx3 Only) 8
New Software Features – PROPixx 9
New Sequencers Added to MATLAB 9
New Software Features – PyPixx Application 10
UI Style Update 10
Search Bar 11
New Demo for Most Devices 13
Digital I/O 13
RESPONSEPixx Tester 16
Register Read features color and new style 17
Stability Improvement of test pattern launcher 17
Pupil calibration (refer to previous section) 18
TRACKPixx Calibration Utility new UI 18
TRACKPixx Calibration Utility has new distance measure 21

Overview

This update guide provides installation and usage information relating to the latest VPixx Software Tool release.

For technical questions or product support information, do not hesitate to contact the VPixx support team by phone or by sending an E-mail to support@vpixx.com

By creating your *MyVPixx* account on the VPixx Technologies website, you will have access to additional product documentation, demos, source code examples and the latest firmware and software drivers.

New Firmware Updates

PROPixx, revision 41

- Limited to bug fixes.

VIEWPixx / PROPixx Controller revision 51

- Limited to bug fixes.

TRACKPixx3, revision 11

- Built-in Pupil calibration
- IR Security is applied on start-up
- Bug fixes and eye-tracking improvements

DATAPixx3, revision 12

- Console automatically detected, stability fixes
- Test Pattern can now be shown on Out 2
- Digital Input can now be set to an output if special update is applied (contact VPixx)
- TRACKPixx Overlay border shows state of the tracker

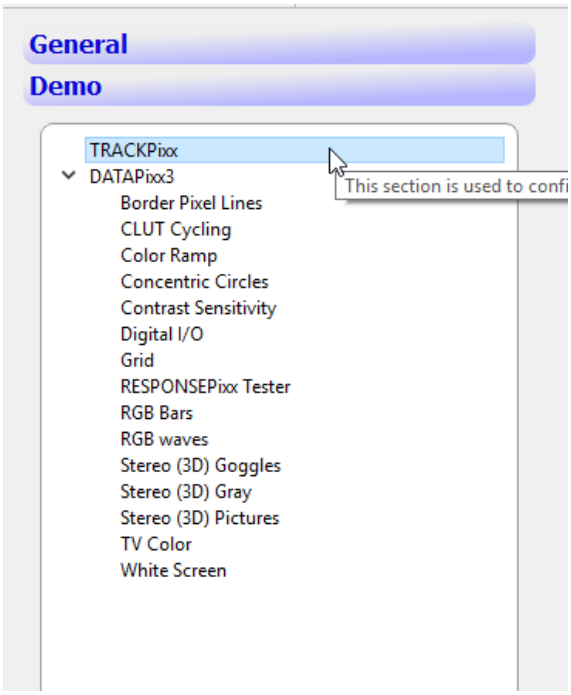
New Software Features – Eye Tracking

Pupil size calibration (TRACKPixx3 only)

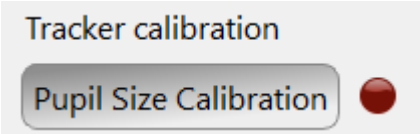
The TRACKPixx3 is the first eye tracker in the world to have this feature. Scientific papers point to the flaws in reported gaze positions given by video-based tracker systems depending on pupillary responses. With this pupil size calibration feature, the gaze position is corrected to reflect pupil constriction or dilation.

PyPixx Implementation

Access the TRACKPixx3 calibration widget by clicking on the **Demo** section and selecting *TRACKPixx3*.



In the TRACKPixx widget, click on *Pupil Size Calibration*.



The red or green light to the right of the button indicates if the TRACKPixx3 calibration has been done. You can reset a current calibration by starting a new one or by clicking CTRL+ALT on your keyboard before clicking to clear the current calibration.

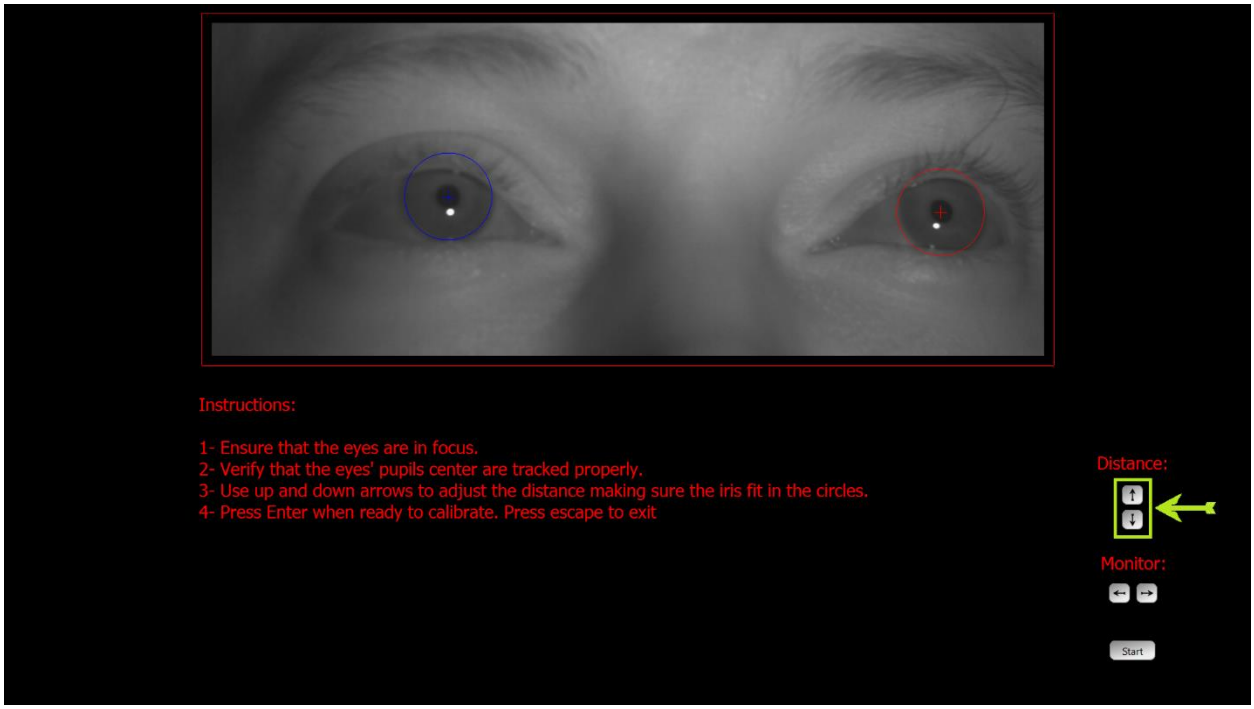
After clicking the button, a full screen display will open on the selected “Stimuli” monitor (in the options). Follow the on-screen instructions.

Distance

The distance settings help adjust the circles to the size of the iris.

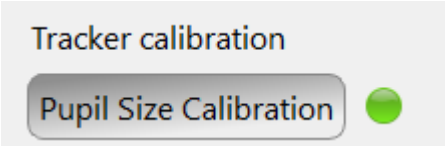
Monitor

You can change the monitor position by using the left and right arrow buttons.



Start the calibration by clicking *Enter* or by pressing the *Start* button. A fixation point of variable brightness will be displayed in the middle of the screen. This should cause a response in pupil size to be used during pupil calibration.

Once the calibration is completed, the widget closes. You can verify that the calibration was successful if the light to the right of the button is green.



MATLAB Implementation

This script has a standalone version which is also implemented in the normal calibration testing script.

Standalone: *TPxPpSizeCal.m*

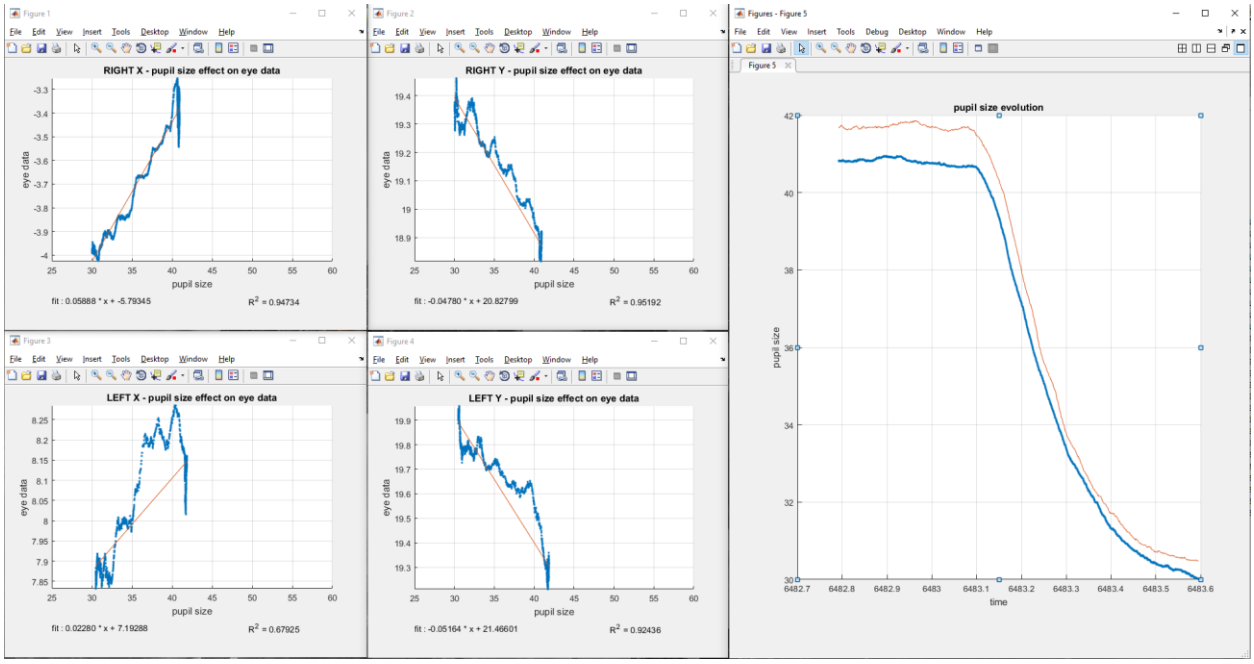
Complete: *TPxTrackpixx3CalibrationTesting.m* (which is a shortcut to call *TPxCalibrationTesting.m*).

The TPxPpSizeCal stand-alone script can be started with no argument. To call it from another script, you must provide the screen pointer as the first argument (usually called windowPtr). The second argument is required if you have not set up the TRACKPixx3 in your current script.

Note: make sure you adjust the ExpectedIrisWidth to the correct pixel value if you are not in a tabletop setup.

Follow the on-screen instructions. A fixation point of variable brightness will be displayed in the middle of the screen. This should cause a response in pupil size, which will be factored in during pupil calibration.

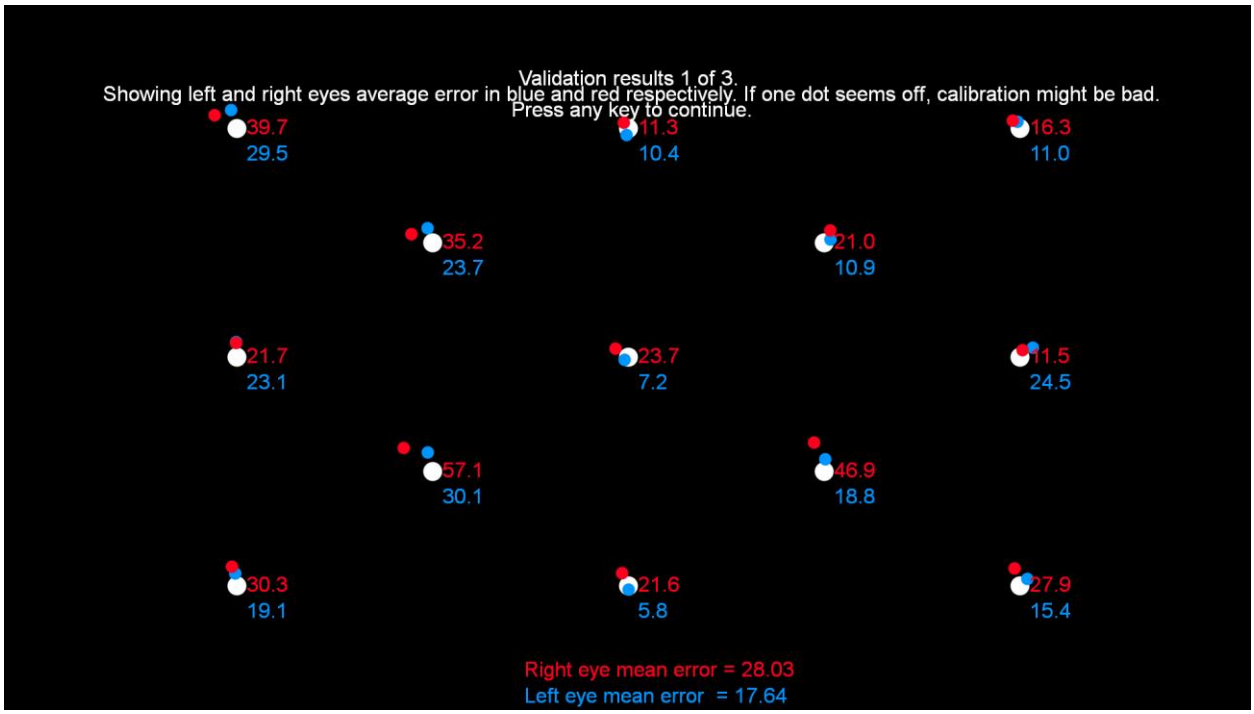
Once the calibration is finished, graphs will be created displaying pupil size according to brightness, which can help you verify that the calibration was successful.



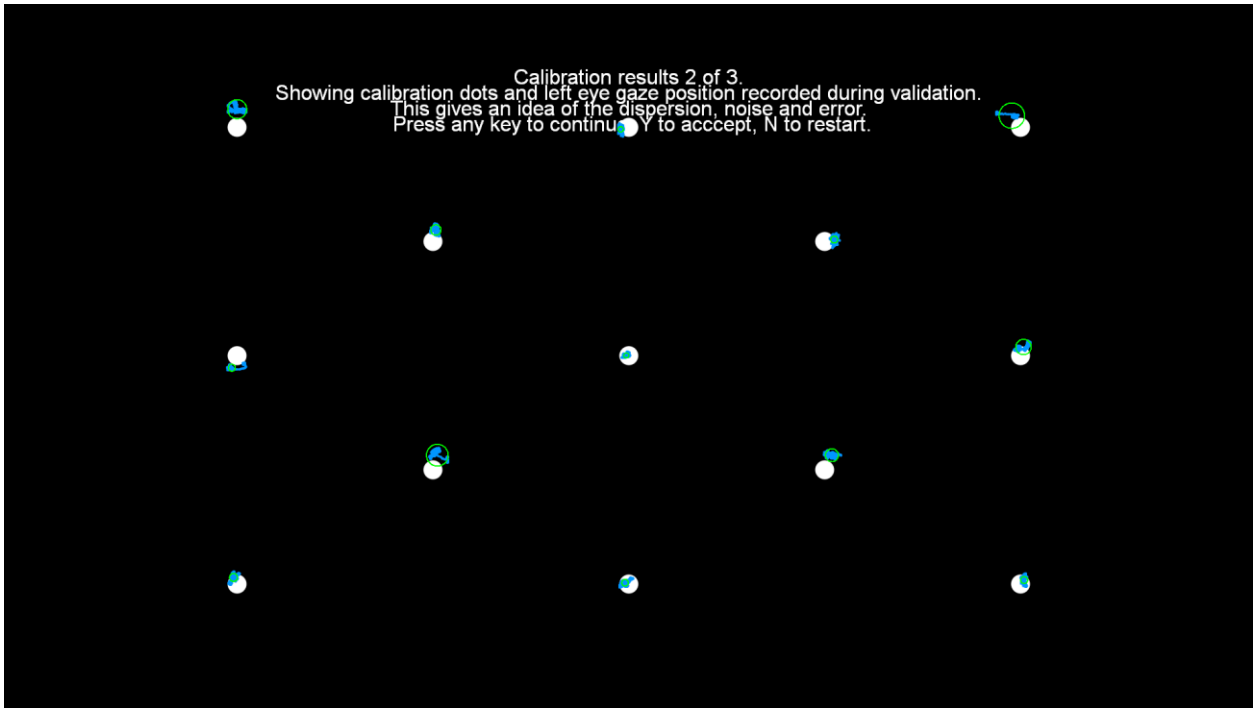
Calibration Validation Script (Matlab) (TRACKPixx3 and TRACKPixx /mini)

The calibration validation script allows you to validate the quality of a calibration for the current setup. It will present the same points as the calibration and gather a huge amount of points for every target. Once the points have finished being presented, the script will display three result screens.

Results 1 of 3: These are the average results for the left and right eye (in screen pixels). If you want to translate this into degrees of visual angle, you will need the distance between the user and the screen as well as the screen size. With this information, simply construct a triangle to find the total visual angle for 1920 pixels and convert the result for the obtained pixel values.



Results 2 of 3 and 3 of 3: These results present all the data gathered during the presentation of the validation points. A circle is centered on the average value. The radius of the circle is equal to two standard deviations for all the data gathered for this point. The smaller the circle, the more precise the data.



TRACKPixx /mini data access (new function and speed improvement)

TRACKPixx /mini data can now be accessed similarly to the TRACKPixx3. A new function in MATLAB called Datapixx('ReadTPxMiniData') has been implemented. This function will return much faster than before (on average, in 2 to 8 ms instead of the previous 8 to 24 ms), allowing for 120 Hz access to the eye-tracking data.

The function can be called as such:

[bufferData] = Datapixx('ReadTPxMiniData');

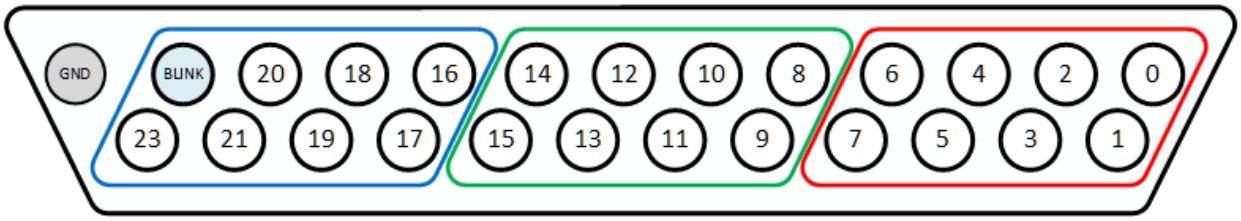
The data returned will be as follows:

[timetag, Left Eye: X, Y, Pupil Size, Right Eye: X, Y, Pupil Size, Distance from TRACKPixx /mini]

Blink detection digital output mode (TRACKPixx3 Only)

When using a TRACKPixx3, it is now possible to enable one of the digital outputs ("Dout" 22, pin 12) to be controlled by the blink detection algorithm. The Dout 22 will be high as soon as one of the eyes is no longer tracked (a situation which is generally detected as a blink).

Here is a diagram of the digital outputs to locate each digital output:



New Software Features – PROPixx

New Sequencers Added to MATLAB

The Datapixx('SetPropixxDlpSequenceProgram' [, program=0]) function now has a few new options that were normally only available as automatically selected in the default mode.

Here is the current list:

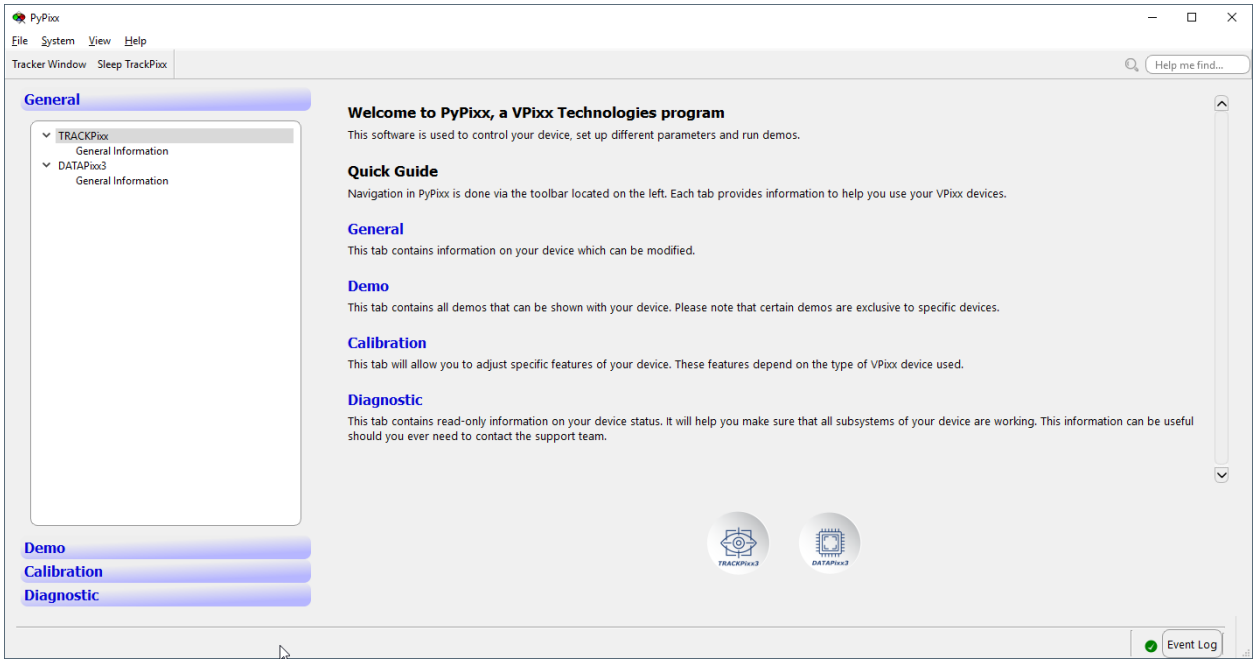
Sequencer Number	Sequencer Name	Definition
0	RGB	Normal Video Processing
1	RB3D	Red/Blue channels contain left/right-eye stereoscopic greyscale images. This mode only supports a refresh rate 120Hz or less.
2	QUAD4X	4 display quadrants are projected at 4x refresh rates, up to 480 Hz. Frame order is: TopLeft, TopRight, BottomLeft, BottomRight.
3	RGB240	Standard sequencer used to generate 240 Hz refresh rate.
4	RGB180	Standard sequencer used to generate 180 Hz refresh rate.
5	QUAD12X	4 display quadrants x3 RGB channels are projected greyscale at 12x refresh rates, up to 1440 Hz. Frame order is: RedTopLeft, RedTopRight, RedBottomLeft, RedBottomRight.
6	RGBHBD	Normal Video Processing, Optimized for high bit depth
7	RGB2	An older version of the standard video processing
9	GREY3X	Converts 640x1080@360 Hz RGB video to 1920x1080@720 Hz greyscale video by using each color information as one pixel at 360 Hz and adding a blank frame in between to create the 720 Hz
10	GREY720	Uses 1280x720@240 Hz RGB to sequentially display the red, green and blue frames in greyscale at 720 Hz.

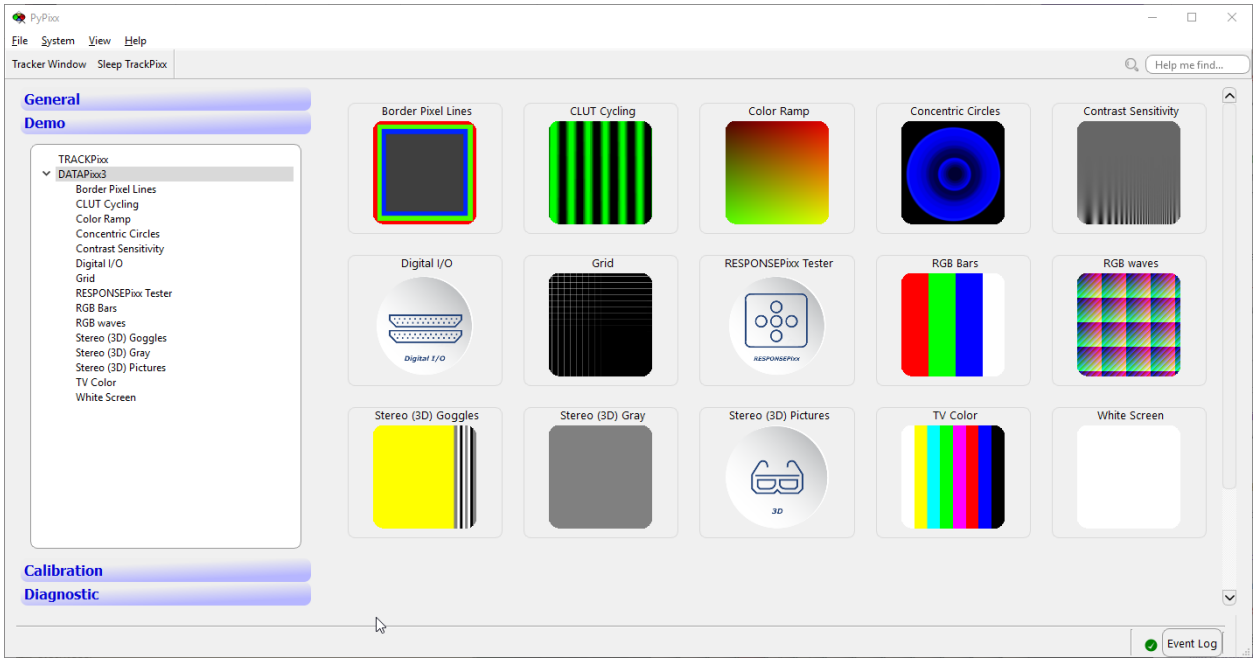
New Software Features – PyPixx Application

UI Style Update

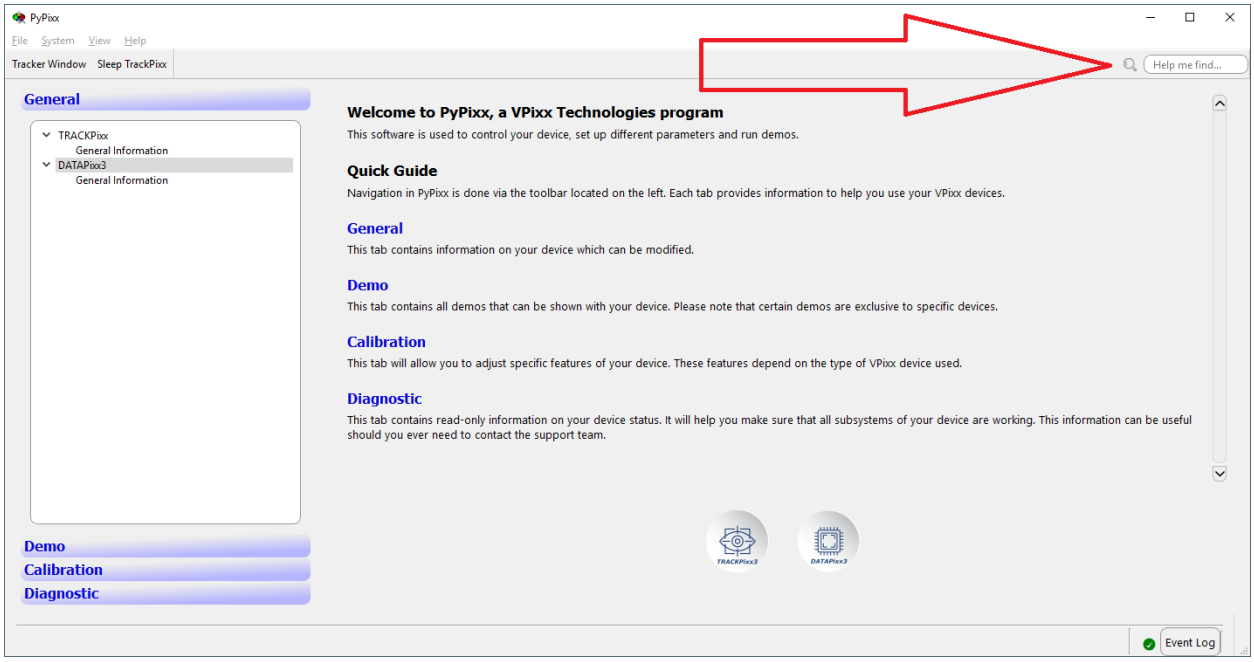
The PyPixx interface has received a complete overhaul.

Opening Screen:



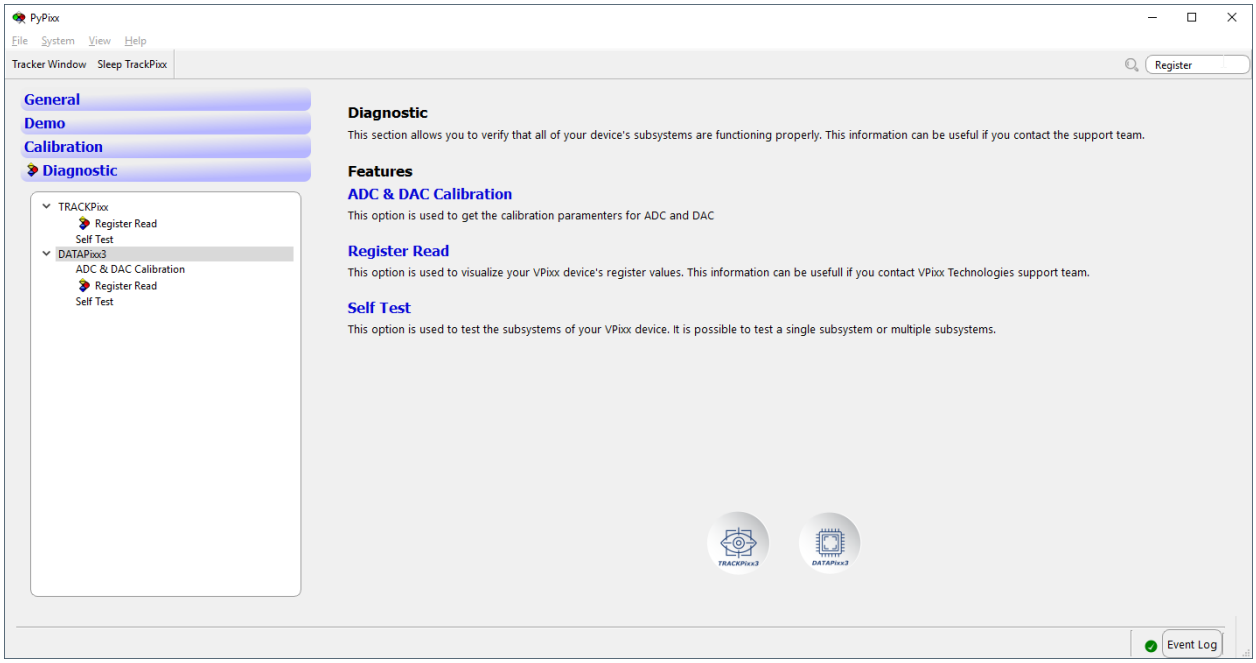


Search Bar

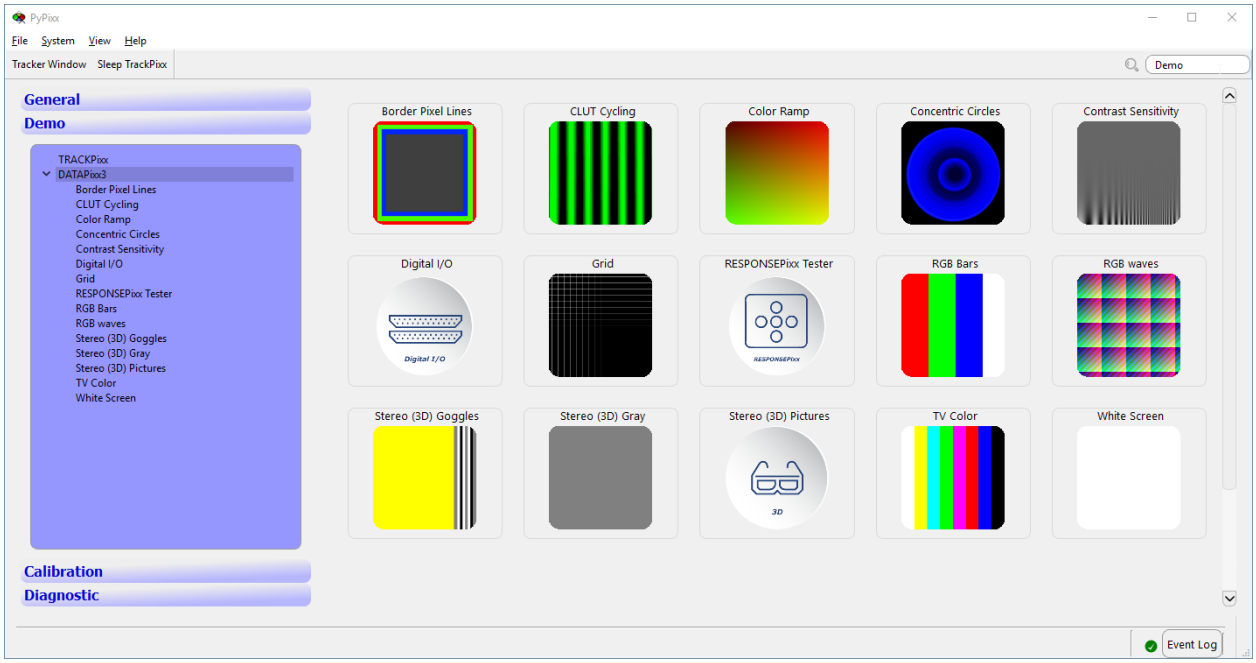


The search bar can help you locate most PyPixx features, with the term you are searching for being highlighted.

For example, if searching for the word ‘Register’:



Another example if you are looking for ‘Demo’:

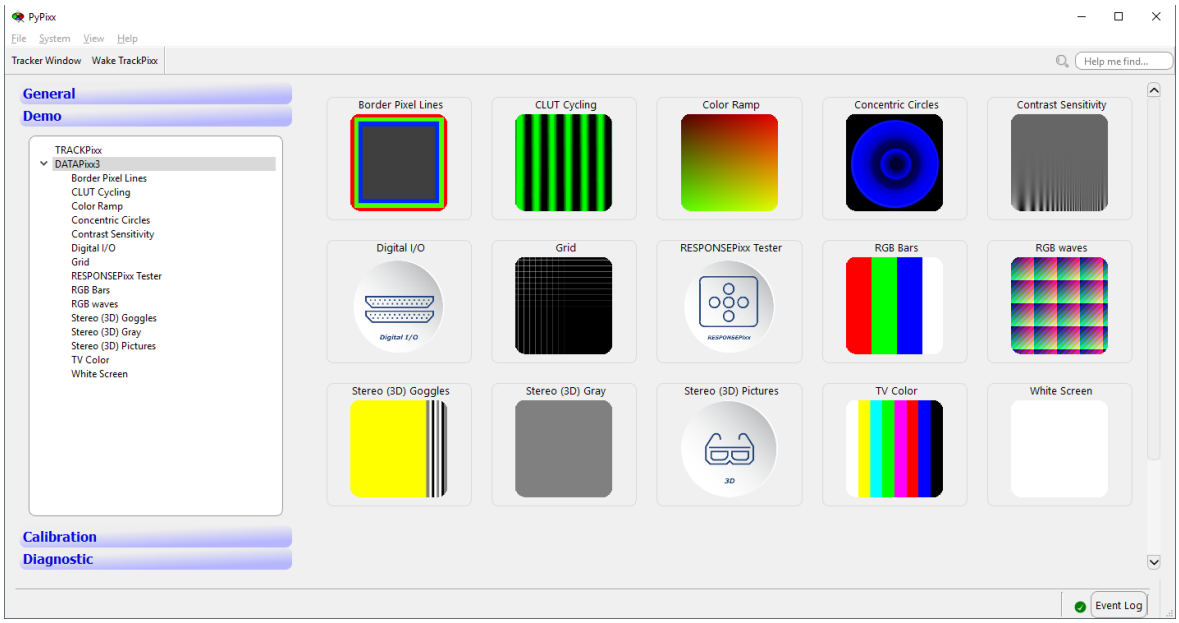


There are also shortcuts for specific commands. This is the list of supported commands:

Register read: -rr
Test Patterns: -tp 1, -tp 5, -tp 9, -tp 10, -tp 12, -tp 13, -tp 17, -tp 18, -tp 19, -tp 29, -tp 23, -tp 27,
Test bench (Diagnostic Tools): -tb
Fimrware update (general information): -fpga

New Demo for Most Devices

The **Demo** Widget in PyPixx has been updated to a more efficient UI.

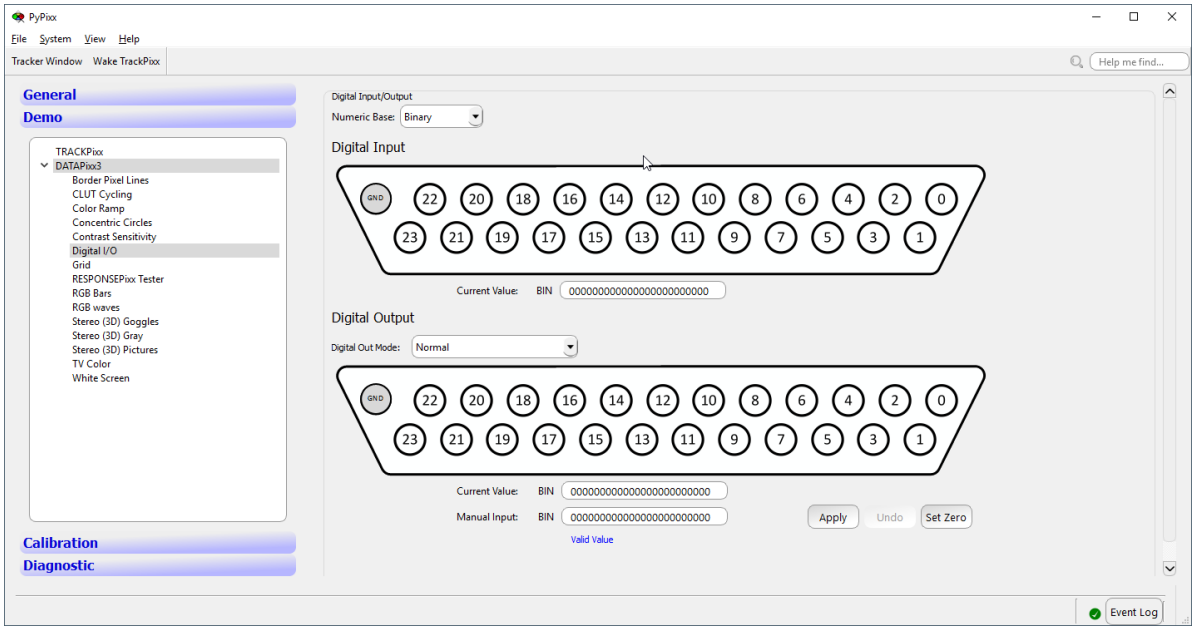


There are two new demos exclusive to PyPixx:

- Digital I/O
- RESPONSEPixx Tester

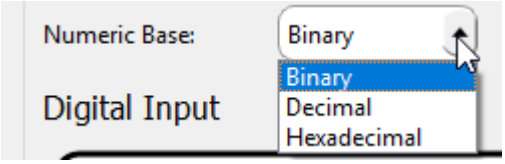
Digital I/O

A widget to test the digital inputs and outputs of devices which have these connectors. You can also set the digital output mode as well as set it to a specific value if you want to test your trigger receiving equipment.



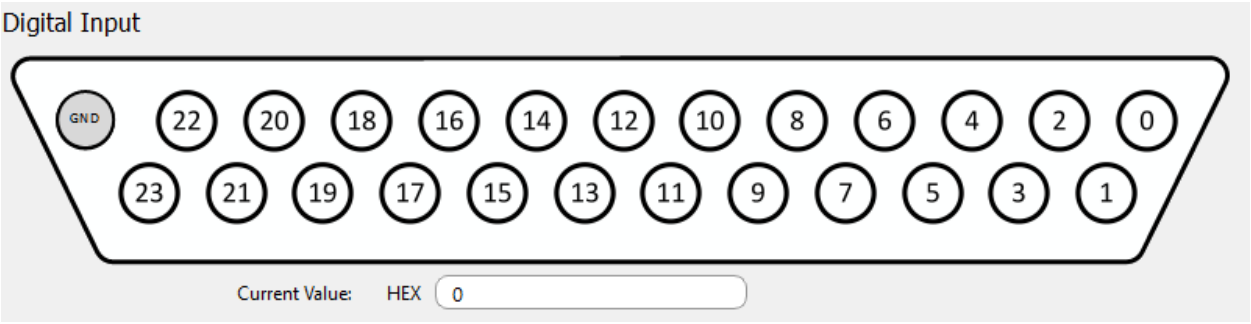
Numeric Base

At the top of the widget is displayed the **Current Device Time** and you can select which numeric base you are using:



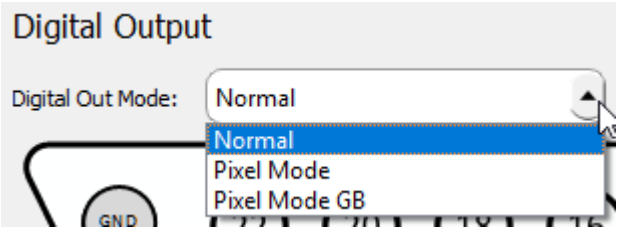
Digital Input

The digital input represents the real time value of the digital inputs. When nothing is plugged in, these are high (green) by default. If you plug another device, such as a REPONSEPixx, these will go low and you can monitor the triggers you receive (if they last long enough to be seen).



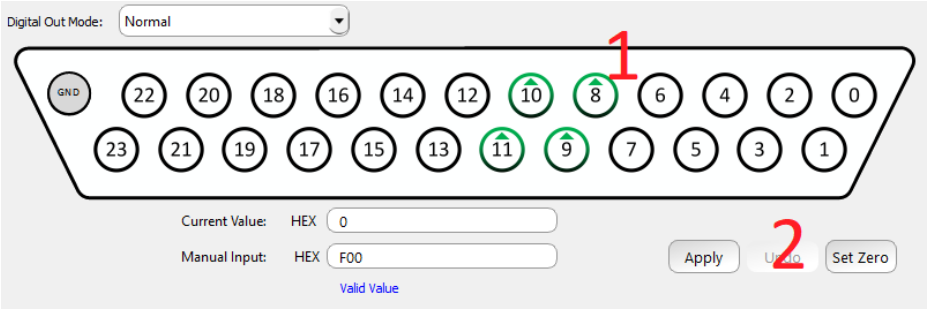
Digital Output

The digital output has 3 modes that can be chosen: Standard, Pixel Mode and Pixel Mode GB.



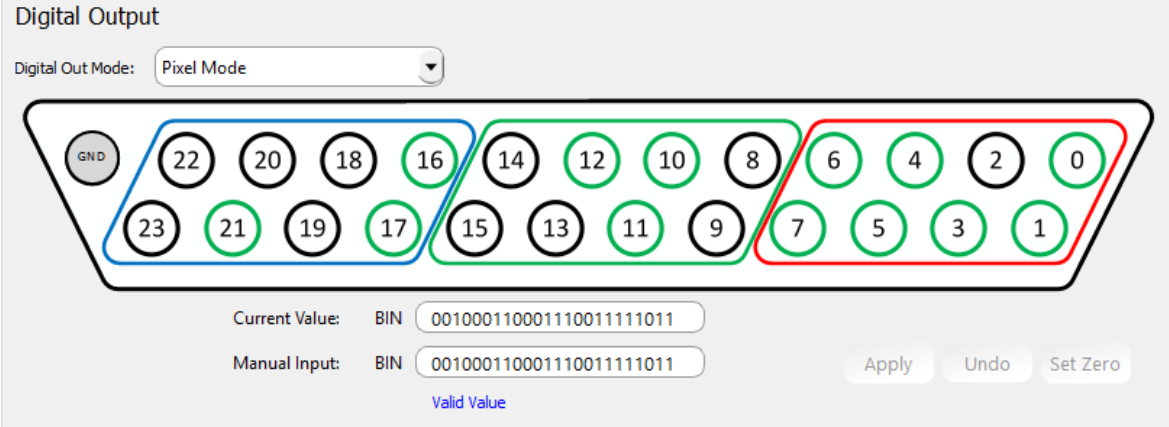
In the normal mode, you can control the digital output in two ways.

You can click each digital output you want to trigger and then press *Apply* to turn them on:

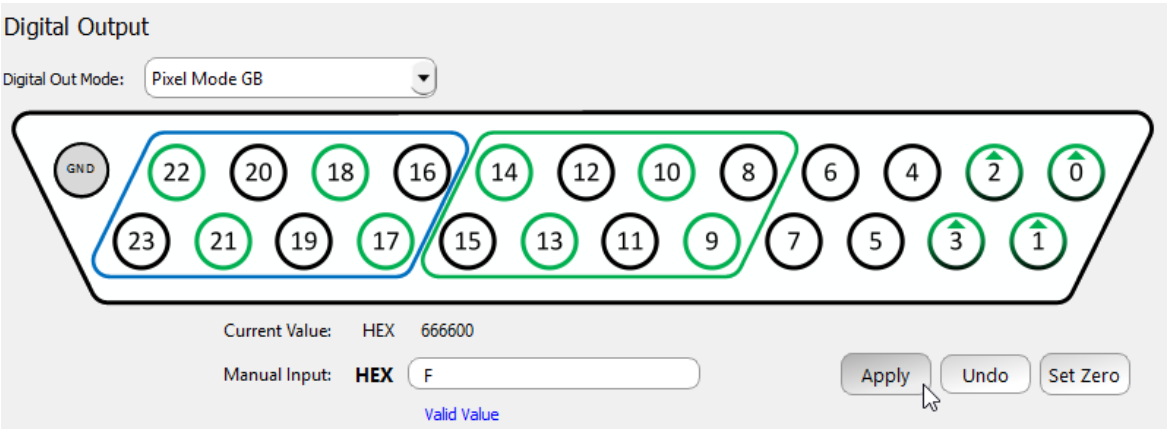


You can also place a manual input at the bottom of the widget in the currently selected numerical base. Again, press *Apply* to change the current digital output value.

In pixel mode, the values of the digital outputs are controlled by the color value of the top left pixel. The first 8 digital outputs are controlled by the red value, the next 8 by the green value and the last 8 by the blue value:

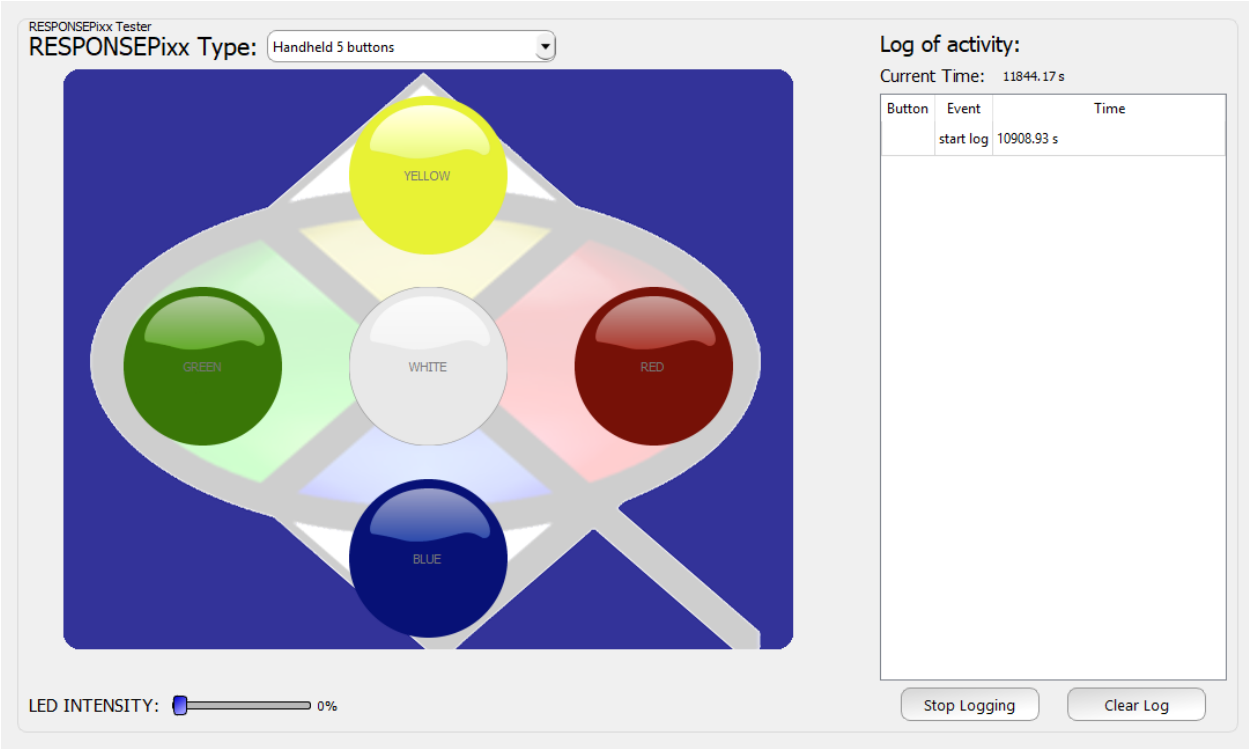


The last mode is pixel mode GB. In pixel mode GB, you can control the lower 8 bits with the normal control (using the manual input and the clicks), while the top 16 bits are controlled by the green and blue color:

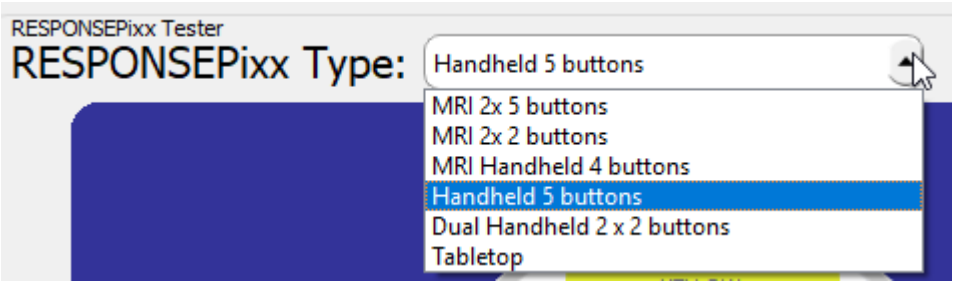


There are two more buttons at the bottom left: *Undo* and *Set Zero*. Undo returns you back to your previous value of digital output while Set Zero will set all the 24 digital outputs to zero (unless they are driven by pixel mode).

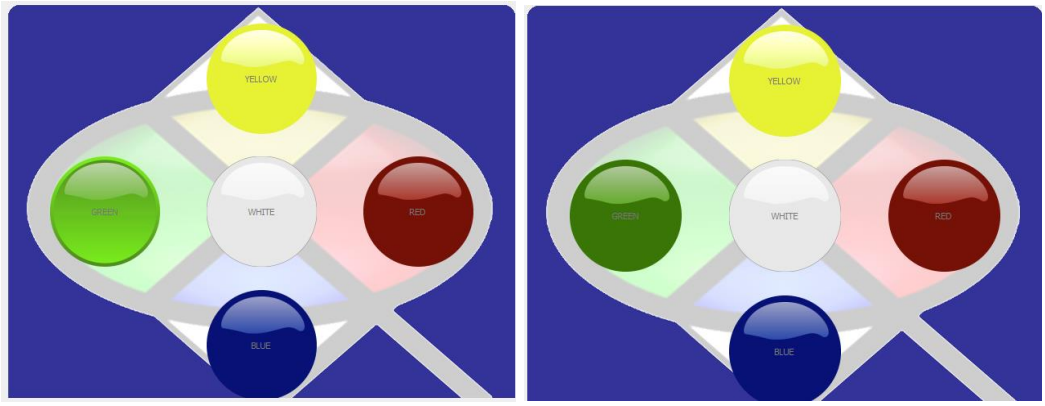
RESPONSEPixx Tester



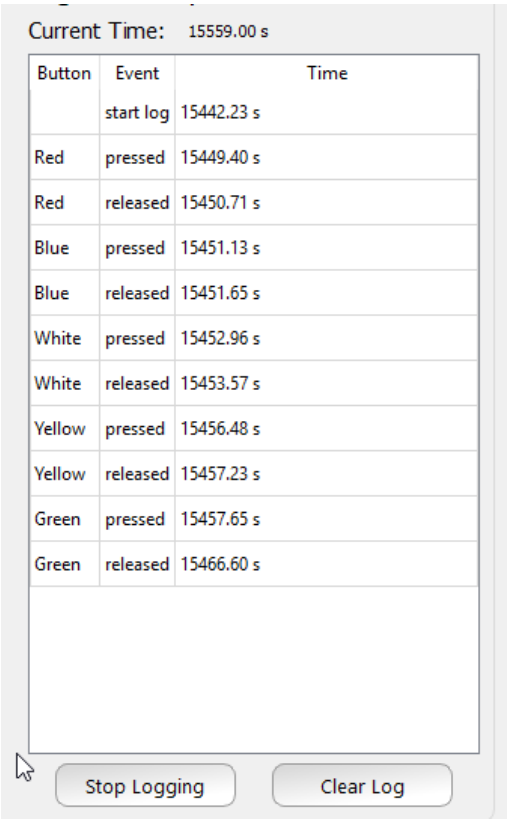
The RESPONSEPixx Tester allows you to test the RESPONSEPixx. At the top of the screen, you can select your RESPONSEPixx Type, depending on the equipment you have plugged in:



With your RESPONSEPixx type selected, you can see that whenever you press a button, the middle of the widget will reflect the state of the RESPONSEPixx (buttons light up when pressed).



On the right side you can see the log of the current button press for this session. This technique is very similar to how you should record button presses in your experiments.



At the bottom of the log there are two buttons, one that stops this feature and one to clear the current frame.

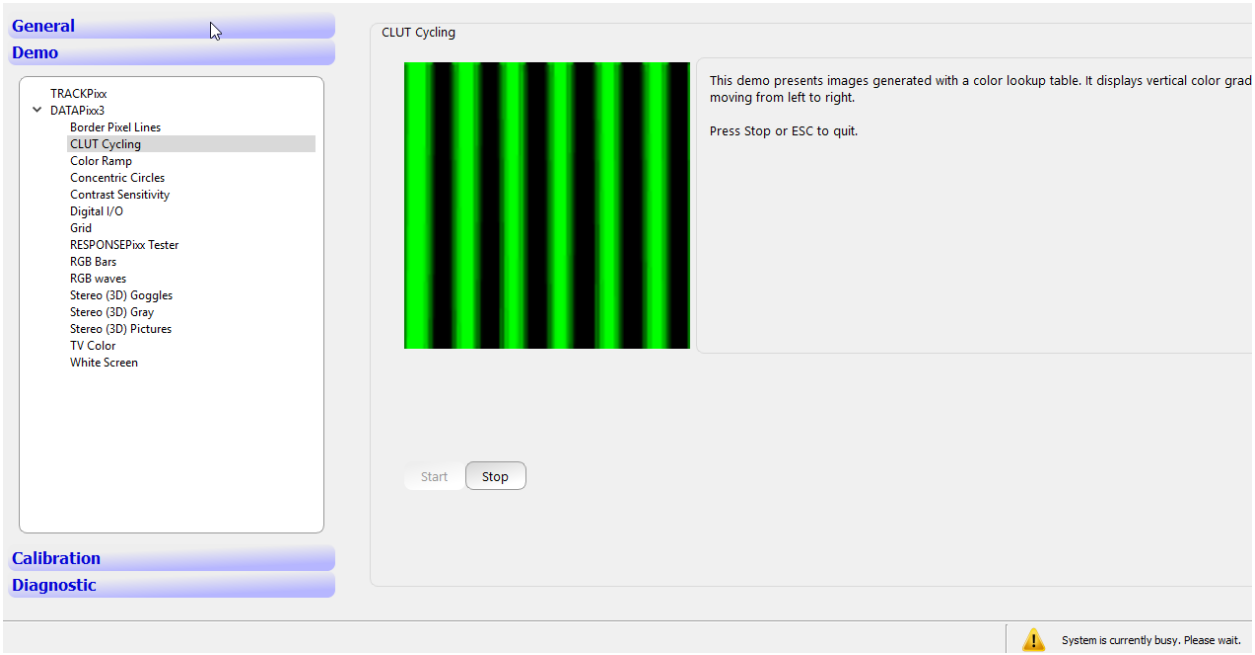
On the bottom left of the widget, there is a slider which allows the non-MRI RESPONSEPixx to have their LEDs turned on. This slider will apply to all LEDs.

Register Read features color and new style

Register read will now feature color for important user information, such as revision number, device name, input resolution, etc.

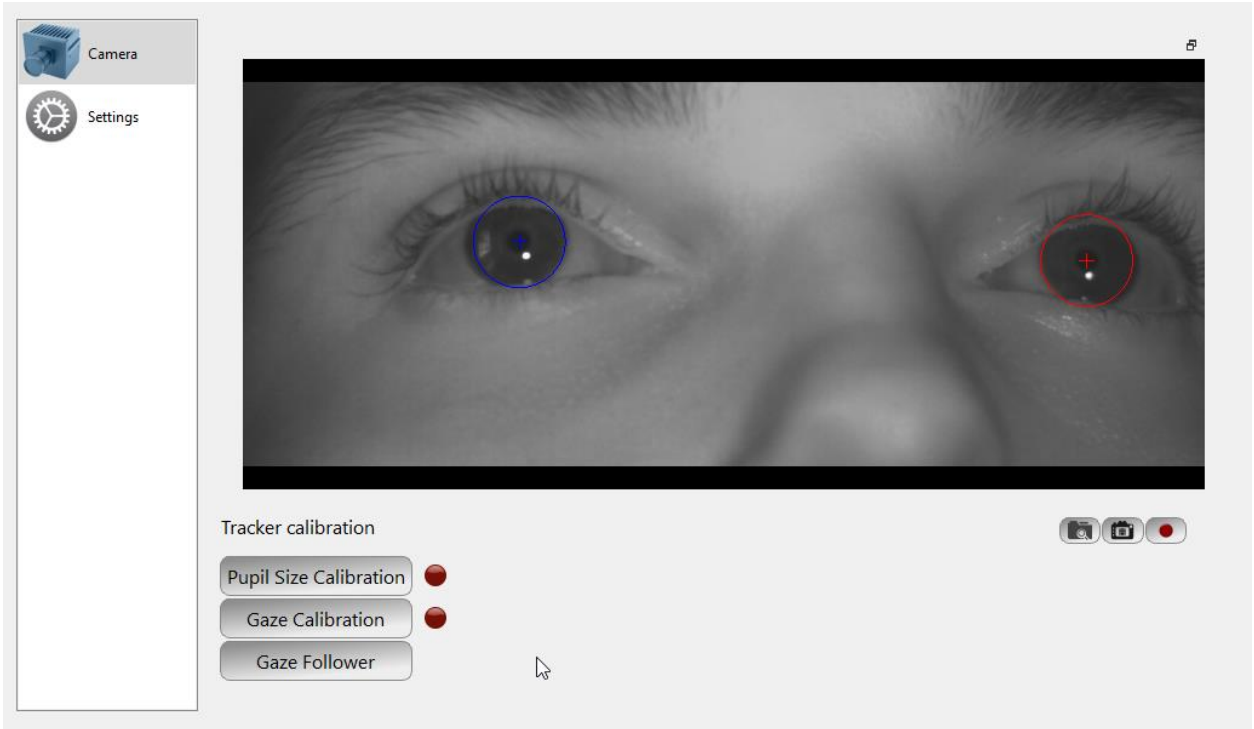
Stability Improvement of test pattern launcher

Alternating between demos and other PyPixx features should now be stable. Previously, errors might have been encountered when switching from a currently running test pattern to another PyPixx feature. Some demos take up to 5-6 seconds to end, which will cause the UI to “freeze” during this time. In a future update, we will offer a loading screen but for now this delay is unavoidable.

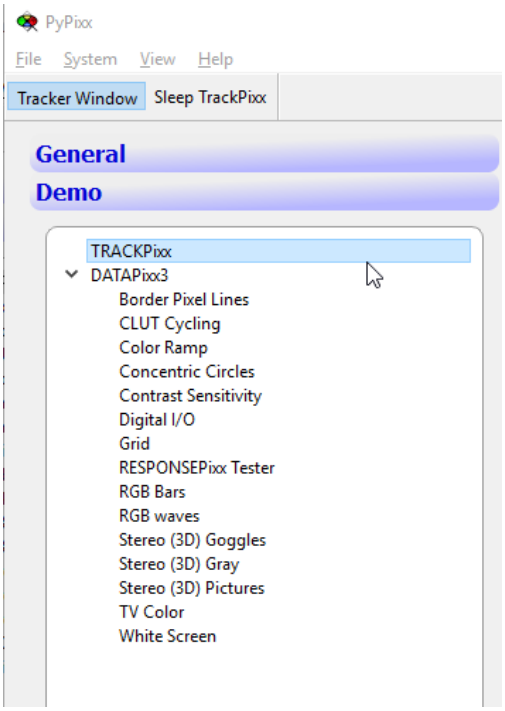


Pupil calibration (refer to previous section)

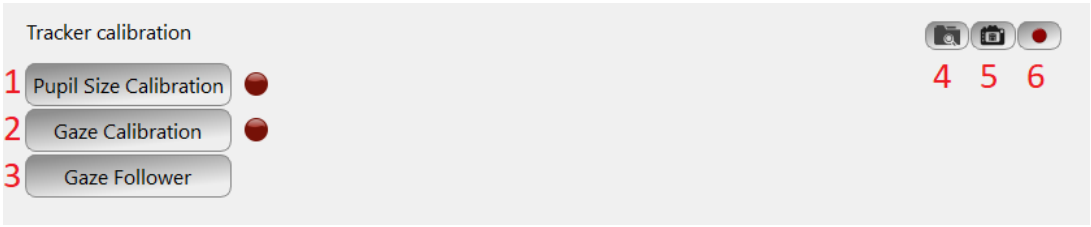
TRACKPixx Calibration Utility new UI



This calibration widget is still accessible within the demo by clicking *TRACKPixx*.

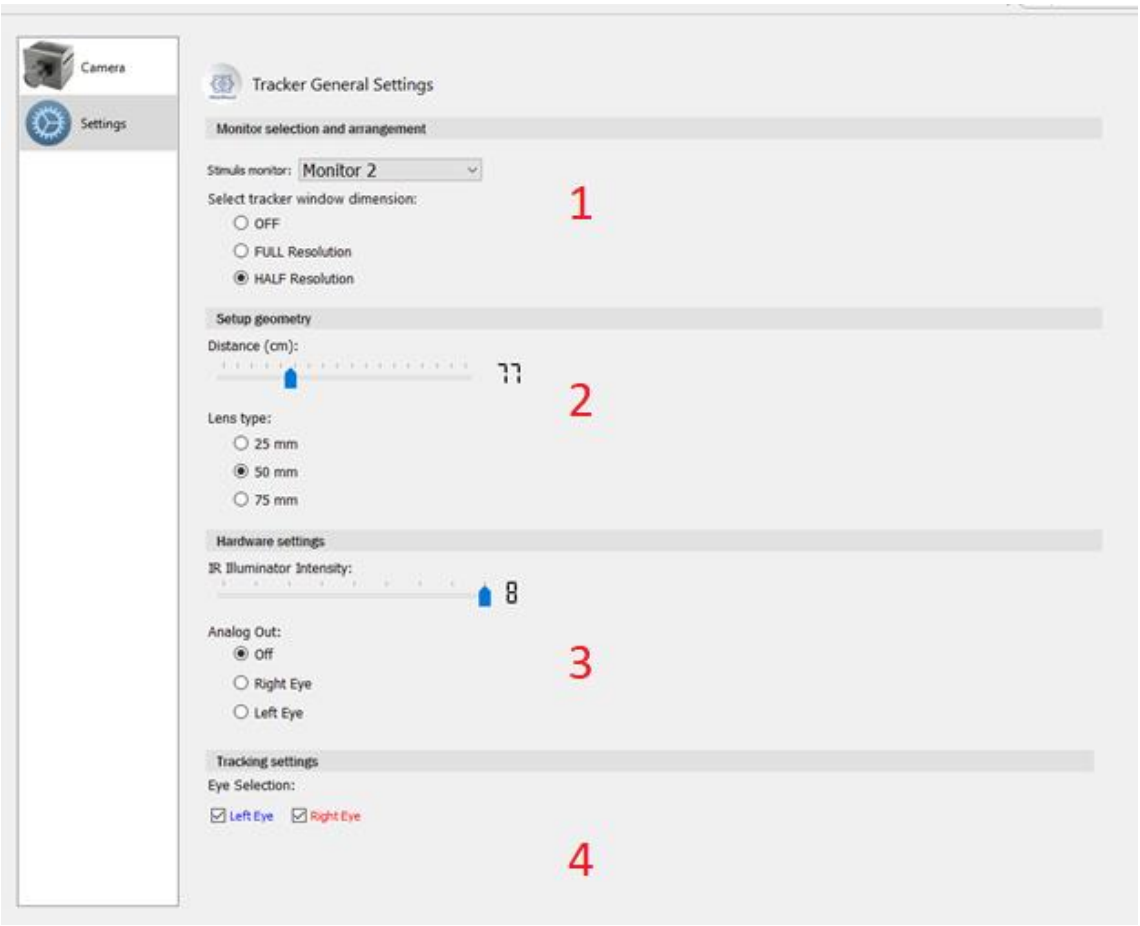


This new widget features 6 buttons.



1) Starts the Pupil Size Calibration (refer to previous section). The light on the right side turns green once calibrated.
2) Starts the standard Gaze Calibration (refer to Application guide). The light on the right side turns green once calibrated.
3) Starts the gaze follower. Requires that gaze calibration be completed.
4) Opens the folder where the data and pictures are saved (see buttons 5 and 6).
5) Takes a picture of the current camera view. Useful if you have issues tracking a certain subject and wish to send camera information for support.
6) Starts recording data. Once started, an indicator on the right of the button shows the recording status:

The new **Settings** widget is as follows:



1) **Monitor selection and arrangement**
   This selects the monitor on which to do the calibration and where the gaze follower will be presented. The tracker window is the overlay shown on the monitor connected in the 'Out 2' of the DATAPixx3. This can be turned on at half or full resolution.

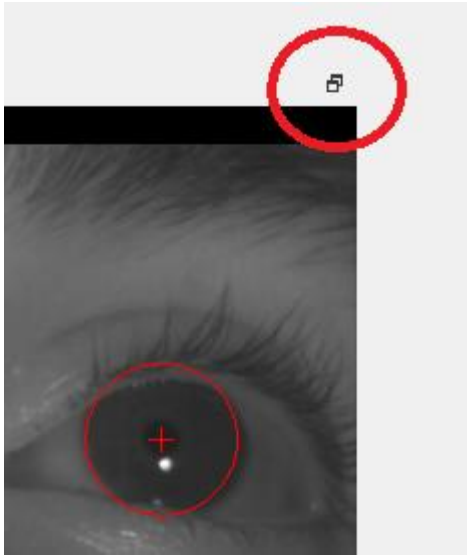
2) **Setup geometry**
   Refer to the next chapter (*TRACKPixx Calibration Utility has new distance measure)*

3) **Hardware settings**
   The InfraRed Illuminator is generally set to maximum power, but should the TRACKPixx be used closer than normal, it might be necessary to reduce its intensity for proper tracking. The Analog Out features can be enabled here. Refer to the Application Guide for more information on the TRACKPixx.

4) **Eye Selection**
   Activating this setting will make it so that one of the eyes will be ignored during calibration. For example, if one eye is not trackable, disabling this eye will speed up the calibration process.

TRACKPixx Calibration Utility has new distance measure

The *Setting* and *Calibration* widgets have a new setting called **Distance** (expressed in centimeters). This setting will influence the size of the circles on the camera image. **The circles should match the size of the iris**. This distance value is an approximation calculated for a normal geometry. You should first select the right lens and then start with the physical distance from the lens to the eyes. To adjust this in the *Settings* widget you can pop out the camera using the button at the top.



Double-click on the camera widget to dock it back in. **This is an important step to perform before calibration, else calibration may not be successful.**